IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Wingyu Leung and Fu-Chieh Hsu

Assignee: Monolithic System Technology, Inc.

Title: Error Detection/Correction Method

Serial No.: 10/800,382 Filed: 3/11/2004

Examiner: Joseph D. Torres Group No.: 2112

Docket No: MST-1898-22D

Date: March 26, 2008

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This Appeal Brief, filed in triplicate, is in support of the Notice of Appeal dated October 26, 2007.

/

/

/

/

/

/

/

/

/

/

/

/

/

/

INDEX

I. REAL PARTY IN INTEREST. 3
II. RELATED APPEALS AND INTERFERENCES 3
III. STATUS OF CLAIMS. 3
IV. STATUS OF AMENDMENTS. 3
V. SUMMARY OF THE CLAIMED SUBJECT MATTER 4
VI. GROUNDS FOR REJECTION TO BE REVIEWED ON APPEAL . 9
VII. ARGUMENTS 9
1. Claims 1 and 4 are distinguished over Ragle. . . 9
A. Ragle does not teach "storing said data portion and said EDC code portion of each byte of the packet in the memory module" and "reading out said data portion and said EDC code portion of each byte of the packet from said memory module" as recited by Claim 1. . . .10
2. Claim 2 is distinguished over Ragle in view of Brune. 13
3. Claim 3 is distinguished over Ragle. 13
VIII. CLAIMS APPENDIX 15
IX. EVIDENCE APPENDIX 17
X. RELATED PROCEEDINGS APPENDIX 17

I. REAL PARTY IN INTEREST

The real party in interest is the assignee, Monolithic System Technology, Inc., pursuant to the Assignment recorded in the U.S. Patent and Trademark Office on August 2, 1993 on Reel 6627, Frame 0912.

II. RELATED APPEALS AND INTERFERENCES

Based on information and belief, there are no other appeals or interferences that could directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-4 are pending and stand rejected. The rejection of these claims is being appealed. Claims 1-4 are listed in the Claims Appendix.

IV. STATUS OF AMENDMENTS

The Applicant canceled Claims 5-11 in response to the Final Office Action dated June 1, 2007. The Examiner entered these proposed amendment in the Advisory Action dated October 10, 2007. No further amendments have been filed subsequent to the final rejection set forth in the Final Office Action dated June 1, 2007.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

In general, the claimed subject matter relates to a method of performing error detection and correction (EDC). In one embodiment, a data packet includes eight 9-bit bytes. In this embodiment, 8 bits in each byte can be used to carry data. The other bit in each byte can be grouped together to carry the EDC code. As illustrated in Figure 9, for an 8-byte data packet, each byte can be used to carry 8 bits of data and 1 bit of the 8 bit EDC code. The EDC code is distributed among the 8 bytes of the packet. (Specification, paragraph [0079].)

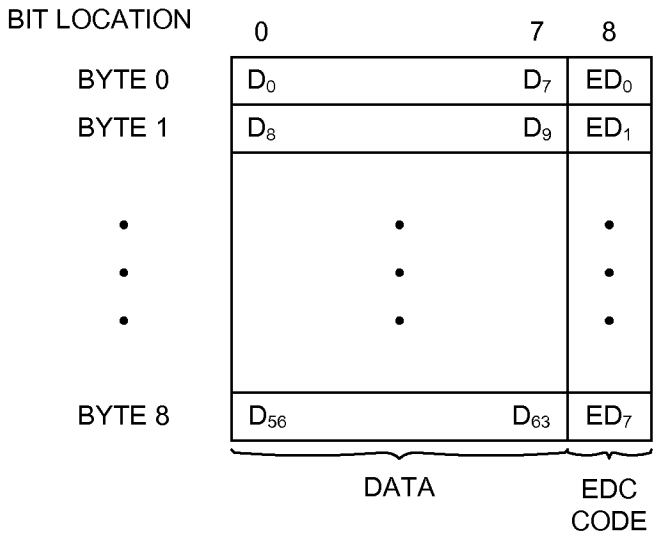

FIG. 9

EDC operations are carried out in the memory controller. Figure 10A shows the block diagram of the memory system using a bus-watch EDC scheme.

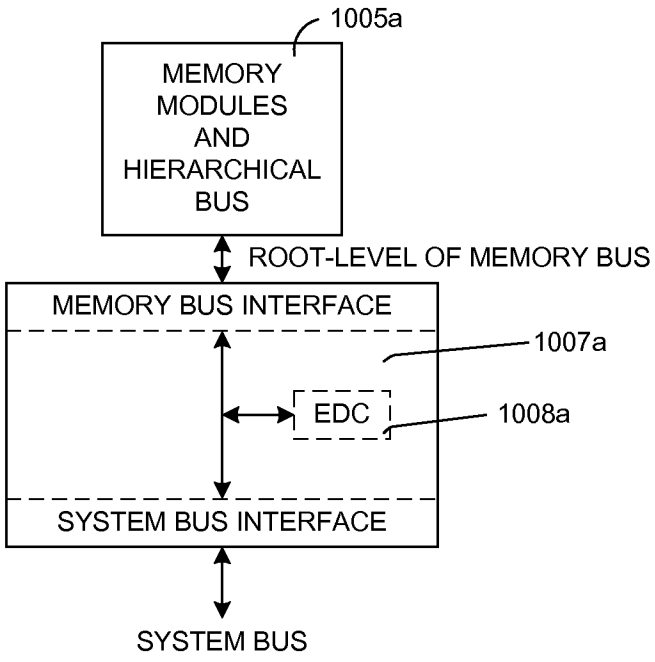

FIG. 10A

During a memory write operation to a memory module (1005a), the memory controller 1007a assembles the data and encodes the EDC code in the data packet before sending it. The destined memory module stores both the EDC code and data indiscriminately, in other words it simply stores the whole packet in the cache or in the memory core of the memory module without further data processing. During a memory read operation, the desired data packet, which contains both the data and its EDC code, is fetched from the memory module 1005a. After arriving at the memory controller 1007a, the EDC bit in each byte is stored away, and the data portion is forwarded to the requesting device in the system. A copy of that data is sent to the EDC functional block 1008a where syndrome bits of the data are generated. Error checking and correction are carried out when the complete EDC code is obtained. In this way, EDC operations are carried out in parallel with data transfer. When no error is detected, as is true most of the time, EDC operations have little effect

on the memory accessing time. When an error is detected, the memory controller 1008a sets a flag in its internal register, corrects the data, writes the correct data back to the memory module, and generates an interrupt to the requesting device to arrange for a data re-transmission. (Specification, paragraph [0080].)

The subject matter of independent Claim 1 finds exemplary support in the specification and drawings as follows:

SUBJECT MATTER	**SPECIFICATION**	**DRAWINGS**
1. A method for error detection and correction (EDC) in transferring data in a packet of bytes from a memory module to a requesting device comprising the steps of:	Paragraph [0080], 'bus-watch EDC scheme'. Paragraph [0079] 'an 8-byte data packet'. Paragraph [0080], 'memory module 1005a'. Paragraph [0080], 'requesting device in the system'.	Figs. 9, 10A
defining each byte of the packet to have an EDC code portion and a data portion, wherein each EDC code portion is a distributed portion of a complete EDC code;	Paragraph [0079], 'each byte can be used to carry 8 bits of data and 1 bit of the 8 bit EDC code. The EDC code is … distributed among the 8 bytes of the packet'.	Fig. 9
storing said data portion and said EDC code portion of each byte of the packet in the memory	Paragraph [0080], 'The destined memory module stores both the EDC code and data indiscriminately, in other words it	Fig. 10A

module;	simply stores the whole packet in the cache or in the memory core without further data processing'.	
reading out said data portion and said EDC code portion of each byte of the packet from said memory module;	Paragraph [0080], 'During a memory read operation, the desired data packet which contains both the data and its EDC code is fetched from the memory module 1005a'.	Fig. 10A
forwarding said data portion of each byte of the packet read from the memory module to said requesting device;	Paragraph [0080], 'After arriving at the memory controller 1007a, … the data portion is forwarded to the requesting device in the system'.	Fig. 10A
storing said EDC portion of each byte of the packet read from the memory module, and sending each said EDC portion to an EDC functional block when the complete EDC code is obtained;	Paragraph [0080], 'After arriving at the memory controller 1007a, the EDC bit in each byte is stored away' … 'Error checking and correction are carried out when the complete EDC code is obtained'.	Fig. 10A
copying said data portion of each byte of the packet read from the memory module, and sending each said data portion to said EDC	Paragraph [0080], 'A copy of that data is sent to the EDC functional block 1008a where syndrome bits of the data are generated'.	Fig. 10A

functional block; and		
performing error checking and correction in said EDC functional block when said EDC functional block receives the complete EDC code.	Paragraph [0080], 'Error checking and correction are carried out when the complete EDC code is obtained'.	Fig. 10A

VI. GROUNDS FOR REJECTION TO BE REVIEWED ON APPEAL

The following rejections are presented to the Board of Appeals for decision:

1. Claims 1 and 4 are rejected under 35 U.S.C. 102(b) as being anticipated by Ragle (U.S. Patent 4,052,698).

2. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ragle in view of Brune (U.S. Patent 3,665,393).

3. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ragle.

VII. ARGUMENTS

1. Claims 1 and 4 are distinguished over Ragle.

Claim 1 recites "defining each byte of the packet to have an EDC code portion and a data portion, wherein each EDC code portion is a distributed portion of a complete EDC code", "storing said data portion and said EDC code portion of each byte of the packet in the memory module" and "reading out said data portion and said EDC code portion of each byte of the packet from said memory module".

Paragraph [0080] of the Application as originally filed specifies an advantage of these steps as follows, "The destined memory module stores both the EDC code and data indiscriminately, in other words it simply stores the whole packet in the cache or in the memory core without further data processing." (Emphasis added.) "When no error is detected as is true most of the time, EDC operations has little effect on the memory accessing time".

A. Ragle does not teach "storing said data portion and said EDC code portion of each byte of the packet in the memory module" and "reading out said data portion and said EDC code portion of each byte of the packet from said memory module" as recited by Claim 1.

Ragle teaches that a parity bit P is computed for each of seven 8-bit characters (i.e., bytes) D1-D7, and that an error check character E is computed for the seven character word, wherein the error check character forms an eighth character of the word. (Ragle, Col. 3, lines 59-68.) However, Ragle teaches that the resulting 8-bit by 9-bit matrix must be converted to a 10-bit by 9-bit matrix before recording onto tape 102. (Ragle, Col. 4, lines 1-4.) The conversion required by Ragle undesirably requires a separate encoder 110. (Ragle, Col. 4, lines 5-23; Figs. 1-3.) The conversion required by Ragle also undesirably requires a larger memory, because the conversion increases the number of required storage bits. The conversion required by Ragle also undesirably increases the memory access time.

However, Ragle teaches that this conversion is required to provide 'no more than two adjacent zeros", and to "never [have] more than one zero leading or ending a code". (Ragle, Col. 4, lines 15-23.) Fig. 2 of Ragle illustrates the conversion of an 8-bit by 9-bit matrix (which includes parity bits P and error check character E) to a 10-bit by 9-bit matrix (which includes generic bits X).

By teaching that this conversion is necessary, Ragle explicitly teaches away from storing the parity bits P and the error check character E on the tape 102. Ragle therefore fails to teach "storing said data portion and said EDC code portion of each byte of the packet in the memory module" as recited by amended Claim 1. Because Ragle fails to teach "storing said data portion and said EDC code

portion of each byte of the packet in the memory module" as recited by amended Claim 1, Ragle also necessarily fails to teach "reading out said data portion and said EDC code portion of each byte of the packet from said memory module" as recited by amended Claim 1.

In rejecting the Applicant's arguments, the Examiner contends that "encoder 110 in Figure 1 of Ragle is a modulation encoder for providing the entire matrix 108 including parity and check bits to tape 102". (Final Office Action, page 3.) This is simply not correct. "The entire matrix 108 including parity and check bits" is an 8x9 array, which is labeled as a "DATA GROUP" in Figure 1 of Ragle. However, this 8x9 DATA GROUP is not written to the tape 102. Instead, Ragle requires that the 8x9 DATA GROUP must be converted into a 9x10 RECORD GROUP that is written to the tape 102. Ragle does not (and cannot) specify the locations of the data, parity and check bits in the converted 9x10 RECORD GROUP. Note that Figure 2 of Ragle shows the converted 9x10 RECORD GROUP with generic bits labeled 'X', while the nature of every bit in the 8x9 DATA GROUP of Fig. 2 is specified as data bit (B), parity bit (P) or check bit (B encircled with dashed lines).

The distinction between bits of the 8x9 DATA GROUP and bits of the 9x10 RECORD GROUP is illustrated by Figure 3 of Ragle. For example, a 4-bit value of '0000' present in the 8x9 DATA GROUP is converted to a 5-bit value '11001' in the 9x10 RECORD GROUP. Suppose that the least significant bit (0000) of the 4-bit value represents a check bit and the three most significant bits (0000) of the 4-bit value represent data bits. Which bits of the converted 5-bit value '11001' represent the original check bit? Which bits of the converted 5-bit value '11001' represent the original data bits? While the converted 5-bit value '11001' may be

representative of the data bits and check bit present in the original 4-bit value, it is clear that the original data bits and check bit are not written to tape 102. By teaching that the original 8x9 DATA GROUP must be converted to the 9x10 RECORD GROUP before being written to the tape 102, Ragle teach away from writing the original 8x9 DATA GROUP to the tape 102. Because the original 8x9 DATA GROUP is not written to the tape 102, the original 8x9 DATA GROUP necessarily cannot be read from the tape 102.

The Examiner has argued that "If the parity and check bits were not stored on tape 102 it would be impossible for demodulation decoder 122 to recover parity and check bits". (Final Office Action, page 3.) This is not true. As described above, the data, parity and check bits of 8x9 DATA GROUP are encoded by encoder 110 to create the 9x10 RECORD GROUP, which is representative of the 8x9 DATA GROUP, but does not include specific data, parity and check bits. The 9x10 RECORD GROUP is written to the tape 102, and is subsequently read from the tape 102 to provide a 9X10 READ GROUP. Decoder 122 retranslates the 9x10 READ GROUP to the 8x9 matrix form. In this manner, "the parity and check bits are not stored on tape 102", and yet it is not "impossible for demodulation decoder 122 to recover parity and check bits" as suggested by the Examiner.

As described above, Ragle fails to teach "storing said data portion and said EDC code portion of each byte of the packet in the memory module" and "reading out said data portion and said EDC code portion of each byte of the packet from said memory module" as recited by Claim 1. For these reasons, Claim 1 is not anticipated by Ragle under 35 U.S.C. 102. Claim 4, which depends from Claim 1, is not anticipated by Ragle for at least the same reasons as Claim 1. The Applicant therefore respectfully requests

reconsideration and withdrawal of the pending rejections of Claims 1 and 4 under 35 U.S.C. 102.

2. Claim 2 is distinguished over Ragle in view of Brune.

Claim 2 has been rejected under 35 U.S.C. 103(a) as being unpatentable over Ragle in view of Brune. Claim 2, which depends from Claim 1, is allowable over Ragle for at least the same reasons as Claim 1. Because Brune does not seem to remedy the above-described deficiencies of Ragle, Claim 2 is allowable over the combination of Ragle and Brune for at least the same reasons that Claim 1 is allowable over Ragle. The Applicant therefore respectfully requests reconsideration and withdrawal of the pending rejection of Claim 2 under 35 U.S.C. 103.

3. Claim 3 is distinguished over Ragle.

Claim 3 has been rejected under 35 U.S.C. 103(a) as being unpatentable over Ragle. As set forth above, independent Claim 1 is allowable over Ragle. Claim 3, which depends from Claim 1, is allowable over Ragle for at least the same reasons as Claim 1. The Applicant therefore respectfully requests reconsideration and withdrawal of the pending rejection of Claim 3 under 35 U.S.C. 103.

For the foregoing reasons, it is submitted that the Examiner's rejections of Claims 1-4 are erroneous, and reversal of these rejections is respectfully requested.

Respectfully submitted,

/E. Eric Hoffman/
E. Eric Hoffman
Attorney for Appellant
Reg. No. 38,186
(925) 895-3545

VIII. CLAIMS APPENDIX

1. (previously amended) A method for error detection and correction (EDC) in transferring data in a packet of bytes from a memory module to a requesting device comprising the steps of:

defining each byte of the packet to have an EDC code portion and a data portion, wherein each EDC code portion is a distributed portion of a complete EDC code;

storing said data portion and said EDC code portion of each byte of the packet in the memory module;

reading out said data portion and said EDC code portion of each byte of the packet from said memory module;

forwarding said data portion of each byte of the packet read from the memory module to said requesting device;

storing said EDC portion of each byte of the packet read from the memory module, and sending each said EDC portion to an EDC functional block when the ~~a~~ complete EDC code is obtained;

copying said data portion of each byte of the packet read from the memory module, and sending each said data portion to said EDC functional block; and

performing error checking and correction in said EDC functional block when said EDC functional block receives the complete EDC code.

2. (previously amended) A method as in claim 1, further comprising the following steps when an error is detected in said EDC functional block:

setting a flag and correcting said data;

writing the correct data back to said memory module; and

generating an interrupt to said requesting device for a later retransmission.

3. (original) A method as in claim 1, wherein each byte of a packet has 8 bits of data and 1 bit of a 8 bit EDC code and said EDC code is distributed among 8 bytes of each packet.

4. (original) A method as in claim 1, wherein said forwarding of said data portion will not begin until an entire packet is received and said entire packet is checked and corrected for error.

IX. EVIDENCE APPENDIX

Not used.

X. RELATED PROCEEDINGS APPENDIX

Not used.